SDTATIC
Software Development Tools and Technology Information Clearinghouse

The DACS Software Development Tools & Technology Information Clearinghouse (SDTATIC):

www.SDTATIC.com

Robert Vienneau
Project Manager
DACS Technical Manager
rob.vienneau@itt.com
315.838.7118

Tom McGibbon, CSDP
DACS Director
tom.mcgibbon@itt.com
315.838.7094

DACS
DoD Data & Analysis Center for Software

SDTATIC
Software Development Tools and Technology Information Clearinghouse

Presentation Agenda

- Purpose
- What is the DACS?
- What is the SDTATIC Clearinghouse?
- SDTATIC Features
- Model Based Development Tools Example
- How You Can Help

- Conference Survey (Q. 1-3)

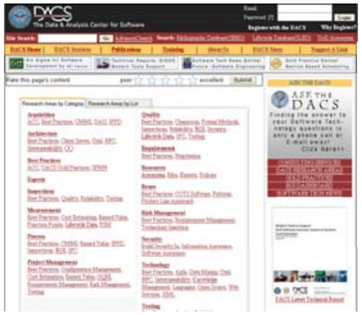
Purpose of This Presentation

- Make you aware of SDTATIC
- Get your Feedback on the Clearinghouse
- Getting you involved
- What Else is Needed?

SDTATIC
Software Development Tools and Technology Information Clearinghouse

DACS - Data & Analysis Center for Software

- ▶ The DACS technical area of focus is Software Technology and Software Engineering, in its broadest sense.
- ▶ Central distribution hub for the latest software technology information sources.
- ▶ Wide variety of Technical Services to support R&D, development, testing, validation, and transitioning of Software Engineering technology.
- ▶ Administered by DTIC. Technically managed by AFRL
- ▶ www.TheDACS.com or iac.dtic.mil/dacs



Modern Tools to Support DoD Software Intensive System of Systems Cost Estimation

THE DACS GOLD PRACTICE INITIATIVE

DACS
DoD Data & Analysis Center for Software

SDTATIC
Software Development Tools and Technology Information Clearinghouse

SDTATIC Clearinghouse

- SDTATIC provides DACS users, staff, Subject Matter Experts (SMEs) with a central and searchable source of information on software development tools and technology.
- At the clearinghouse, users will find a uniform description, characterization, and where available unbiased reviews of software development tools.
- These tools are categorized by a taxonomy
- Initial capability implemented

Software Development Tools

- A software development tool is an executable software product supporting developers during the software system life cycle.
  - A software development tool, as defined here, excludes defined manual techniques, procedures, and processes. It includes commercial as well as open and free tools.
- The focus of SDTATIC is on technology-oriented tools, as opposed to tools for managing and acquiring software.
- SDTATIC Strategy: Prototype with one tool category and expand to other categories

SDTATIC
Software Development Tools and Technology Information Clearinghouse

Sample Categories of Tools

- Architecture Tools
- Requirements
- Design
- Construction
- Testing
- Maintenance
- Open vs. Proprietary

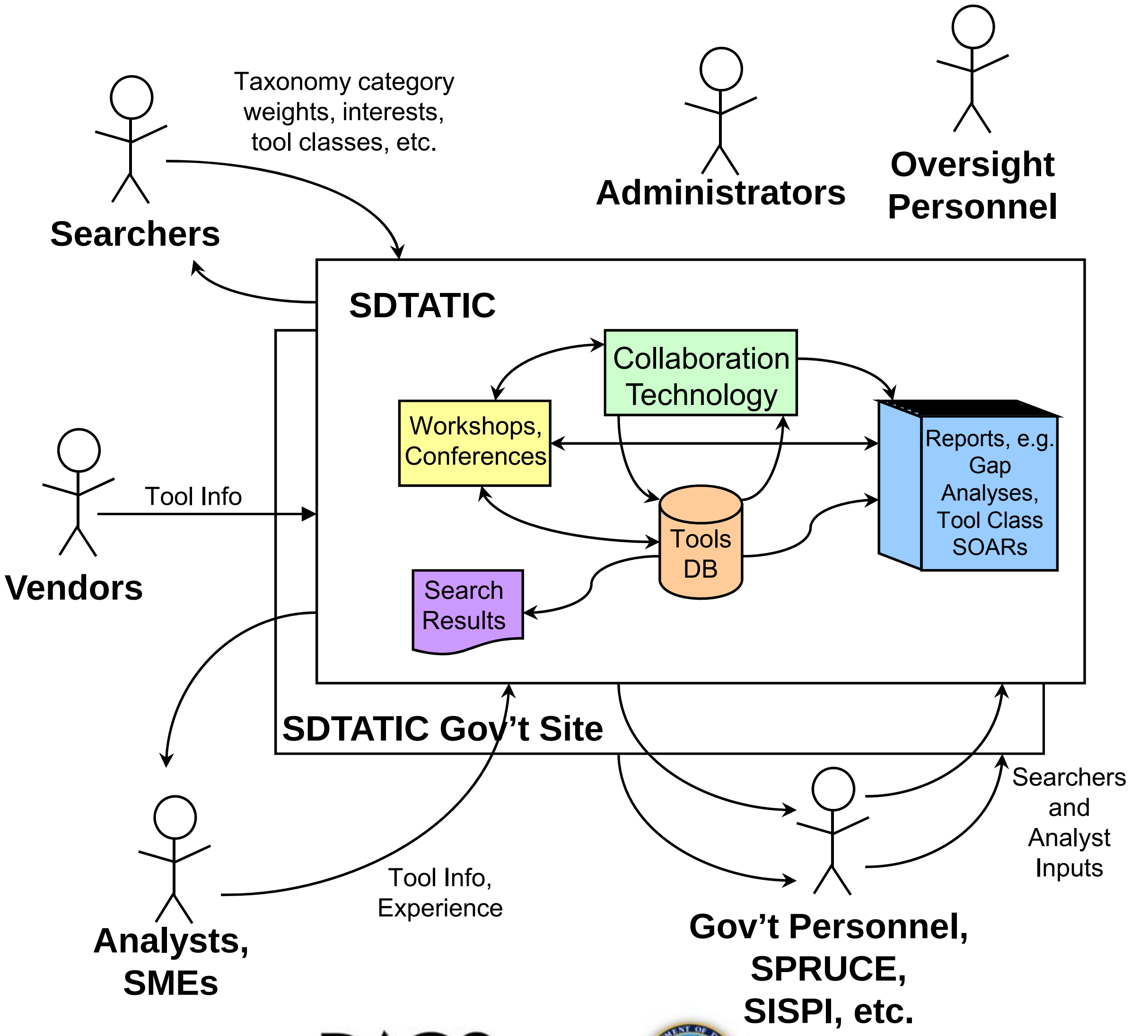
- These are tool attributes

- Embedded Development
- Model Driven Software Engineering
- Software Assurance

DACS
DoD Data & Analysis Center for Software

SDTATIC
Software Development Tools and Technology Information Clearinghouse

SDTATIC Context



SSTC
22 April 2009

DoD Data & Analysis Center for Software

Taxonomy Overview

- Available on SDTATIC web site
- Defined as three-level hierarchy. First level:
  - Life cycle process
  - Functionality
  - Host or running platform
  - Target platform
  - Input type or language
  - Output type or language
  - Availability
- Taxonomy entry includes definition. Maintained wiki-style

SDTATIC
Software Development Tools and Technology Information Clearinghouse

Taxonomy Development

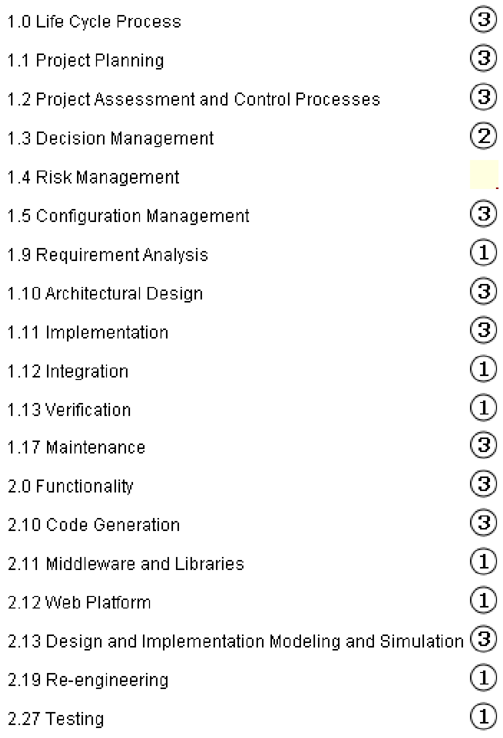
- Synthesizes existing taxonomies
- Life cycle decomposition
  - Based on ISO/IEC 15288:2008(E)
- Functionality from:
  - SWEBOK, Chapter 10
  - INCOSE (for requirements functionality)
- Target Platform
  - Extends Software Development Tools Directory
  - Extensions include Web-based and Middleware

DACS
DoD Data & Analysis Center for Software

SDTATIC
Software Development Tools and Technology Information Clearinghouse

Profiles

- Profiles are associated with the SDTATIC taxonomy
  - Used to prioritize tool requirements
  - Assign an importance to an item in the taxonomy (not important, somewhat important, important, very important)
- Uses:
  - Define what is important to a user
  - Define what is important for a technology area (e.g., testing tools) or other grouping of tools
  - Identify stretch needs for gap analysis



1.0 Life Cycle Process	③
1.1 Project Planning	③
1.2 Project Assessment and Control Processes	③
1.3 Decision Management	②
1.4 Risk Management	
1.5 Configuration Management	③
1.9 Requirement Analysis	①
1.10 Architectural Design	③
1.11 Implementation	③
1.12 Integration	①
1.13 Verification	①
1.17 Maintenance	③
2.0 Functionality	③
2.10 Code Generation	③
2.11 Middleware and Libraries	①
2.12 Web Platform	①
2.13 Design and Implementation Modeling and Simulation	③
2.19 Re-engineering	①
2.27 Testing	①

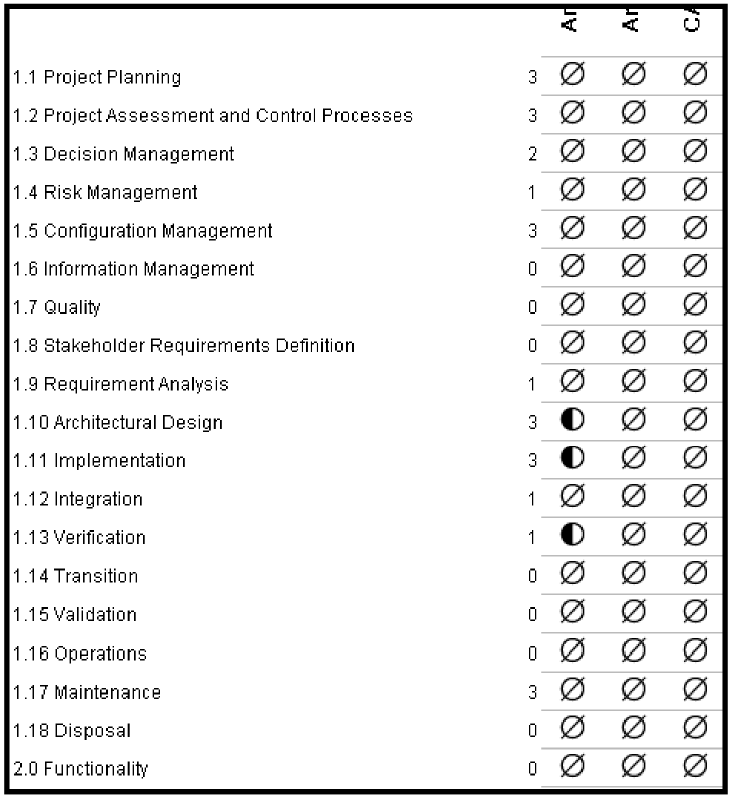
DACS
DoD Data & Analysis Center for Software

SDTATIC
Software Development Tools and Technology Information Clearinghouse

Representing Tools in the Taxonomy

- Tools are evaluated against the taxonomy (not implemented, partially fulfilled, fulfilled)
- DACS will initially develop and maintain assessment
  - Inputs from users welcome
  - Inputs from SMEs welcome
- Side by Side Comparison

- Suggestions: Survey Q4

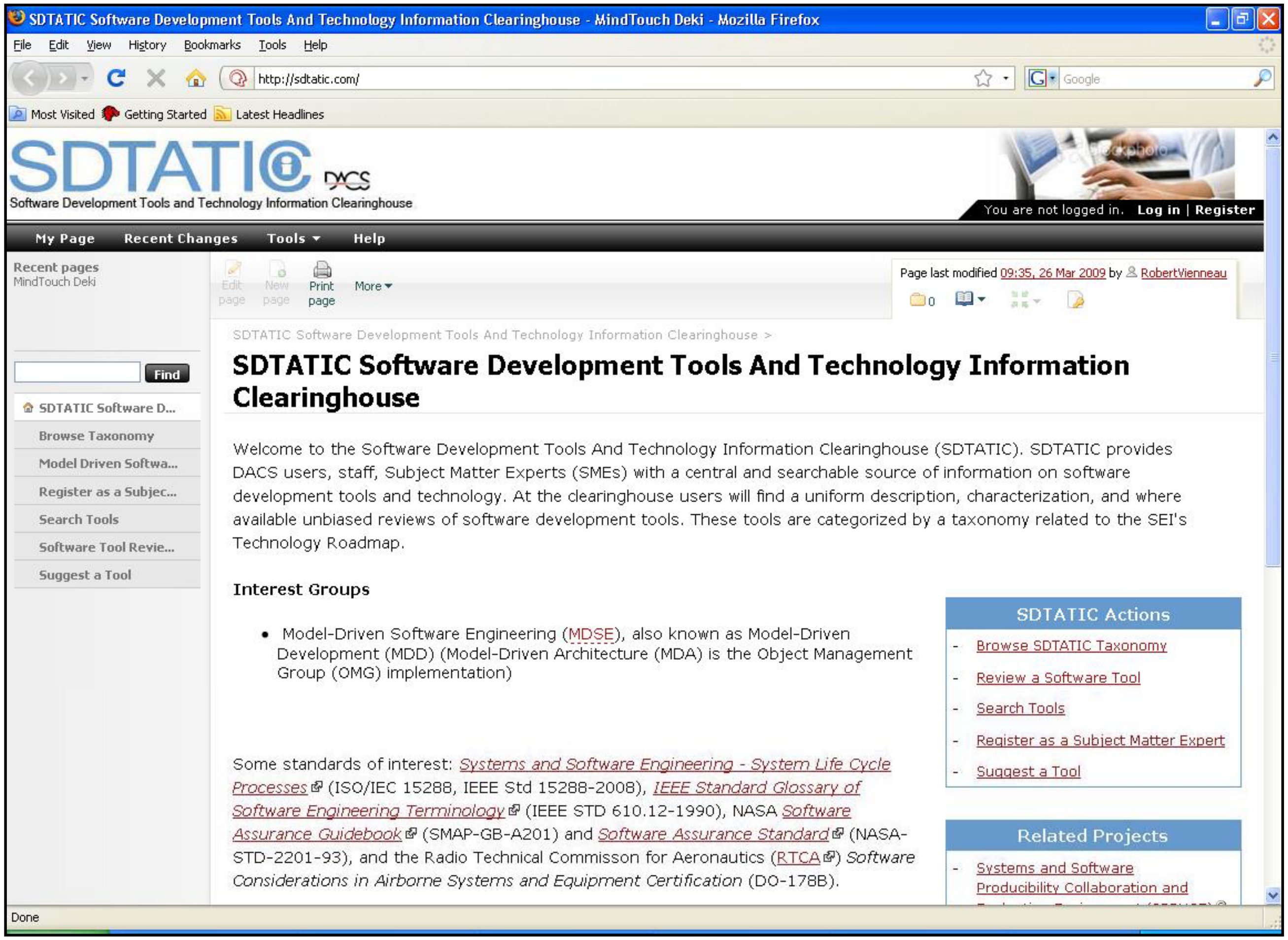


		Ar	Ar	CA
1.1 Project Planning	3	∅	∅	∅
1.2 Project Assessment and Control Processes	3	∅	∅	∅
1.3 Decision Management	2	∅	∅	∅
1.4 Risk Management	1	∅	∅	∅
1.5 Configuration Management	3	∅	∅	∅
1.6 Information Management	0	∅	∅	∅
1.7 Quality	0	∅	∅	∅
1.8 Stakeholder Requirements Definition	0	∅	∅	∅
1.9 Requirement Analysis	1	∅	∅	∅
1.10 Architectural Design	3	◐	∅	∅
1.11 Implementation	3	◐	∅	∅
1.12 Integration	1	∅	∅	∅
1.13 Verification	1	◐	∅	∅
1.14 Transition	0	∅	∅	∅
1.15 Validation	0	∅	∅	∅
1.16 Operations	0	∅	∅	∅
1.17 Maintenance	3	∅	∅	∅
1.18 Disposal	0	∅	∅	∅
2.0 Functionality	0	∅	∅	∅

SDTATIC
Software Development Tools and Technology Information Clearinghouse

The SDTATIC Site

www.SDTATIC.com


SDTATIC Software Development Tools And Technology Information Clearinghouse - MindTouch Deki - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://sdtatic.com/
Google
Most Visited Getting Started Latest Headlines
SDTATIC DACS
Software Development Tools and Technology Information Clearinghouse
You are not logged in. Log in | Register
My Page Recent Changes Tools Help
Recent pages
MindTouch Deki
Find
SDTATIC Software D...
Browse Taxonomy
Model Driven Softwa...
Register as a Subjec...
Search Tools
Software Tool Revie...
Suggest a Tool
Edit page
New page
Print page
More
Page last modified 09:35, 26 Mar 2009 by RobertVienneau
0
SDTATIC Software Development Tools And Technology Information Clearinghouse >
Done

SDTATIC Software Development Tools And Technology Information Clearinghouse

Welcome to the Software Development Tools And Technology Information Clearinghouse (SDTATIC). SDTATIC provides DACS users, staff, Subject Matter Experts (SMEs) with a central and searchable source of information on software development tools and technology. At the clearinghouse users will find a uniform description, characterization, and where available unbiased reviews of software development tools. These tools are categorized by a taxonomy related to the SEI's Technology Roadmap.

Interest Groups

- Model-Driven Software Engineering (MDSE), also known as Model-Driven Development (MDD) (Model-Driven Architecture (MDA) is the Object Management Group (OMG) implementation)

Some standards of interest: *Systems and Software Engineering - System Life Cycle Processes* (ISO/IEC 15288, IEEE Std 15288-2008), *IEEE Standard Glossary of Software Engineering Terminology* (IEEE STD 610.12-1990), NASA *Software Assurance Guidebook* (SMAP-GB-A201) and *Software Assurance Standard* (NASA-STD-2201-93), and the Radio Technical Commisson for Aeronautics (RTCA) *Software Considerations in Airborne Systems and Equipment Certification* (DO-178B).

SDTATIC Actions

- Browse SDTATIC Taxonomy
- Review a Software Tool
- Search Tools
- Register as a Subject Matter Expert
- Suggest a Tool

Related Projects

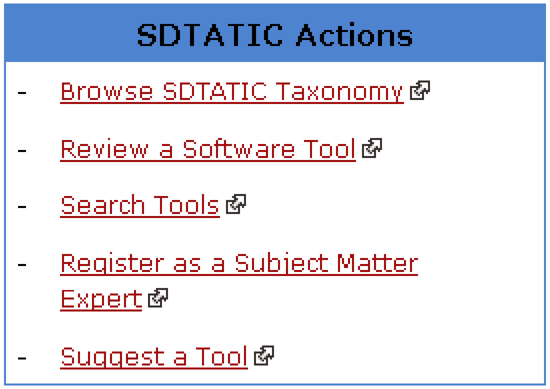
- Systems and Software Producibility Collaboration and

DACS
DoD Data & Analysis Center for Software

SDTATIC
Software Development Tools and Technology Information Clearinghouse

User Capabilities for Finding Tools

- Browsing
- Searching
  - Near term: profile searching
  - Long term: natural language
    - “design tools that generate Java or C++”
- Ranking
  - Weighted rank order of tools based on profile priority
    - Similar to QFD Approach
- Survey Q5



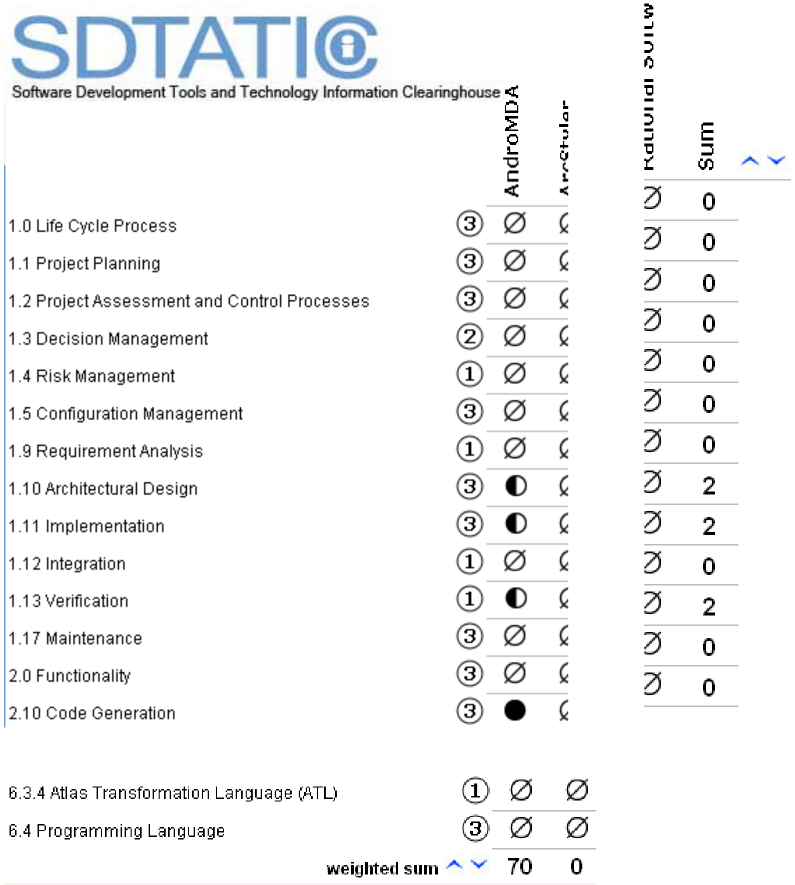
SDTATIC Actions

- Browse SDTATIC Taxonomy
- Review a Software Tool
- Search Tools
- Register as a Subject Matter Expert
- Suggest a Tool

SDTATIC
Software Development Tools and Technology Information Clearinghouse

Finding Technology Gaps

- SDTATIC Gap Analysis Approach based on Quality Function Deployment (QFD)
- Each column for each tool generates a weighted sum.
  - This weighted sum can be used to sort most relevant to least relevant tool
- Each row for each taxonomy category is summed.
  - Totals can be viewed as the extent to which the "market" addresses those features
  - Poorly scored features could be interpreted as "gaps"
- Survey Q6

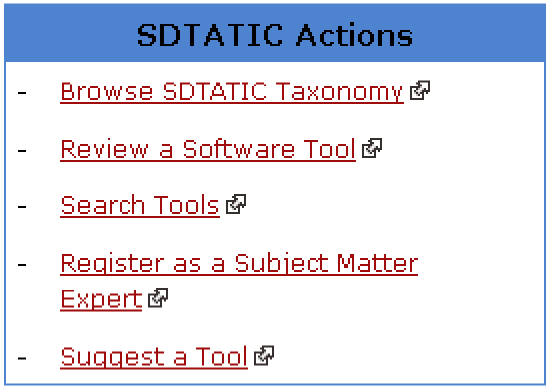


SDTATIC
Software Development Tools and Technology Information Clearinghouse

		AndroMDA	ArcStyler	Rational Software	Sum
1.0 Life Cycle Process	③	∅	∅	∅	0
1.1 Project Planning	③	∅	∅	∅	0
1.2 Project Assessment and Control Processes	③	∅	∅	∅	0
1.3 Decision Management	②	∅	∅	∅	0
1.4 Risk Management	①	∅	∅	∅	0
1.5 Configuration Management	③	∅	∅	∅	0
1.9 Requirement Analysis	①	∅	∅	∅	0
1.10 Architectural Design	③	◐	∅	∅	2
1.11 Implementation	③	◐	∅	∅	2
1.12 Integration	①	∅	∅	∅	0
1.13 Verification	①	◐	∅	∅	2
1.17 Maintenance	③	∅	∅	∅	0
2.0 Functionality	③	∅	∅	∅	0
2.10 Code Generation	③	●	∅		
6.3.4 Atlas Transformation Language (ATL)	①	∅	∅		
6.4 Programming Language	③	∅	∅		
weighted sum		70	0		

Calling all SMEs

- Subject Matter Experts (SMEs) on Tool Technology Areas
- SMEs on Individual Tools
- DACS will work with SMEs for high quality assessments
  - Will contract with selected SMEs
- We will contact you with user questions
  - Provides you direct access to users
- Survey: Q7

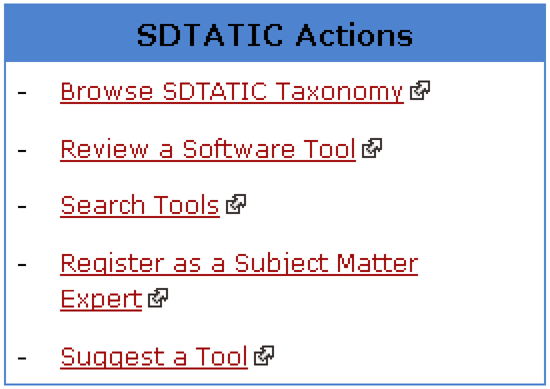


SDTATIC Actions

- Browse SDTATIC Taxonomy
- Review a Software Tool
- Search Tools
- Register as a Subject Matter Expert
- Suggest a Tool

Calling Software Development Tool Vendors

- SDTATIC will collaborate with tool vendors for high quality assessments
  - Tool vendor assessments will be shown separately
- We will either contact you or you can contact us.



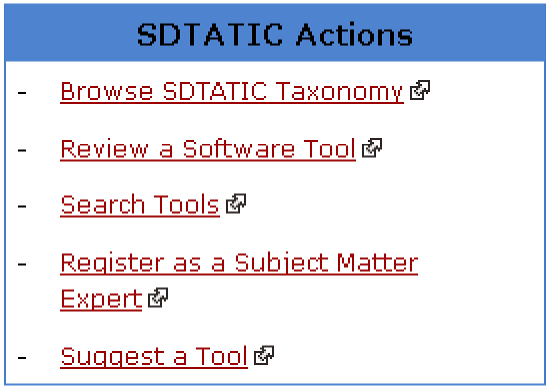
SDTATIC Actions

- Browse SDTATIC Taxonomy
- Review a Software Tool
- Search Tools
- Register as a Subject Matter Expert
- Suggest a Tool

- Survey: Q8 if you are a tool vendor

Getting Your Input

- Capabilities exist to provide inputs/reviews on tools
- SDTATIC.com is a wiki
- SDTATIC Community Building
- Suggest Tools
- Survey: Q9



SDTATIC Actions

- Browse SDTATIC Taxonomy
- Review a Software Tool
- Search Tools
- Register as a Subject Matter Expert
- Suggest a Tool

SDTATIC Community Building

- Work with related projects, e.g.
  - DoD Best Practices Clearinghouse
  - International Council on Systems Engineering (INCOSE)
  - Software Assurance Metrics and Tools Evaluation (SAMATE)
  - Software Systems Stockroom (S3)
  - Systems and software Producibility Collaboration and Evaluation Environment (SPRUCE)
- Use collaborative technology (e.g., wiki)
- Surveys from DACS
- Sponsor workshops, conference tracks, etc.

- Survey: Q10

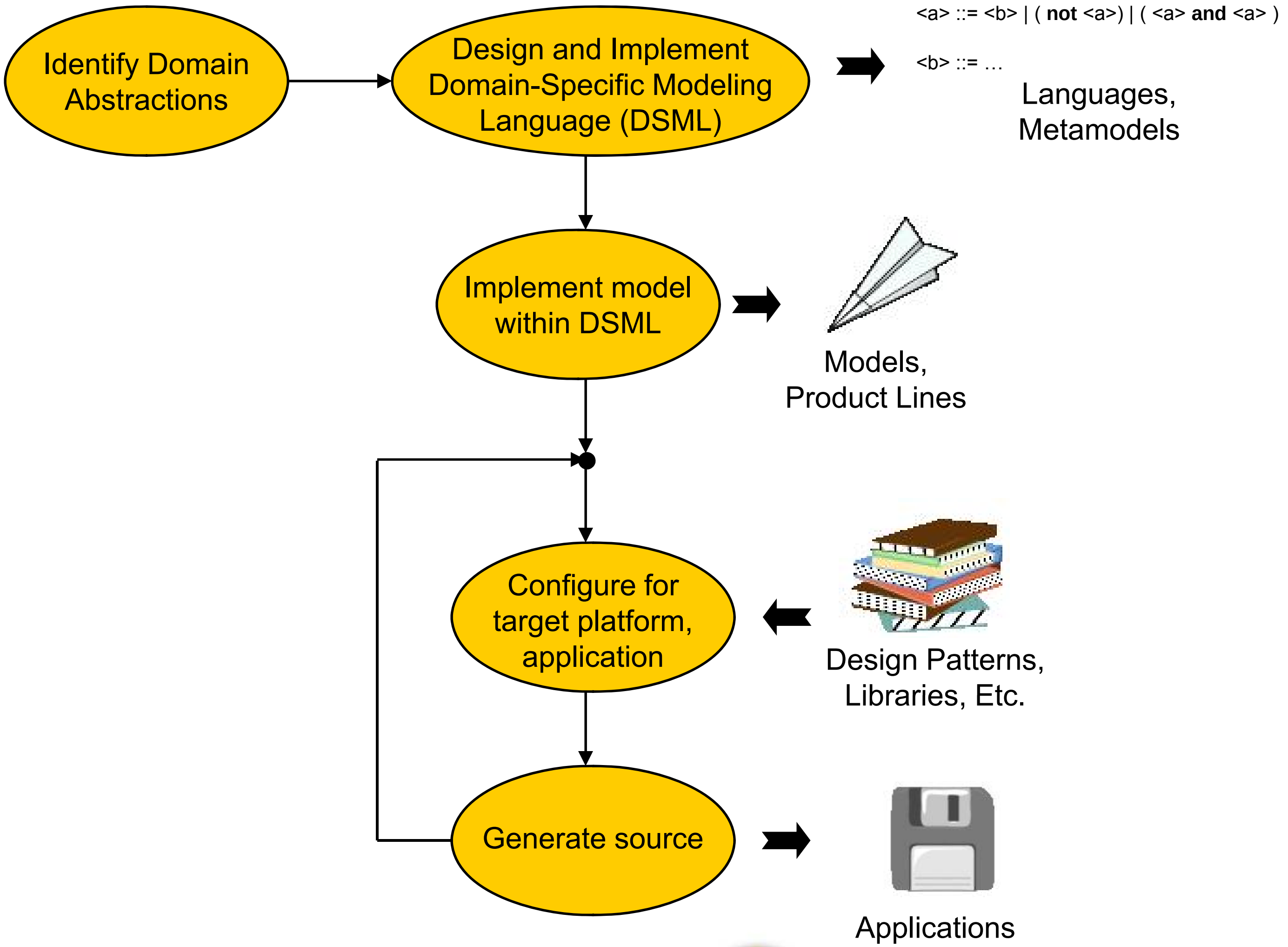
Other Services and Information From SDTATIC

- DACS/SDTATIC Team will Respond to Technical Inquiries on Software Development Tools, up to 4 hours, for Free
- Other Information
  - For Open Source, links to the source
  - Related documents
  - Conference links
  - Vendor links

Model-Driven Software Development

- Definition: Model-driven development is simply the notion that we can construct a model of a system that we then transform into the real thing... A model is a coherent set of formal elements describing something (for example, a system, bank, phone, or train) built for some purpose that is amenable to a particular form of analysis... Model-driven development automates the transformation of models from one form to another. (Mellor et al 2003)
- Synonyms:
  - Model-Driven Architecture (MDA)
  - Model-Driven Development (MDD)
  - Model-Based Development (MBD)
  - Model-Driven Software Engineering (MDSE)

MDD Process



Identify Domain Abstractions

Design and Implement Domain-Specific Modeling Language (DSML)

<a> ::= <b> | (**not** <a>) | (<a> **and** <a>)

<b> ::= …

Languages, Metamodels

Implement model within DSML

Models, Product Lines

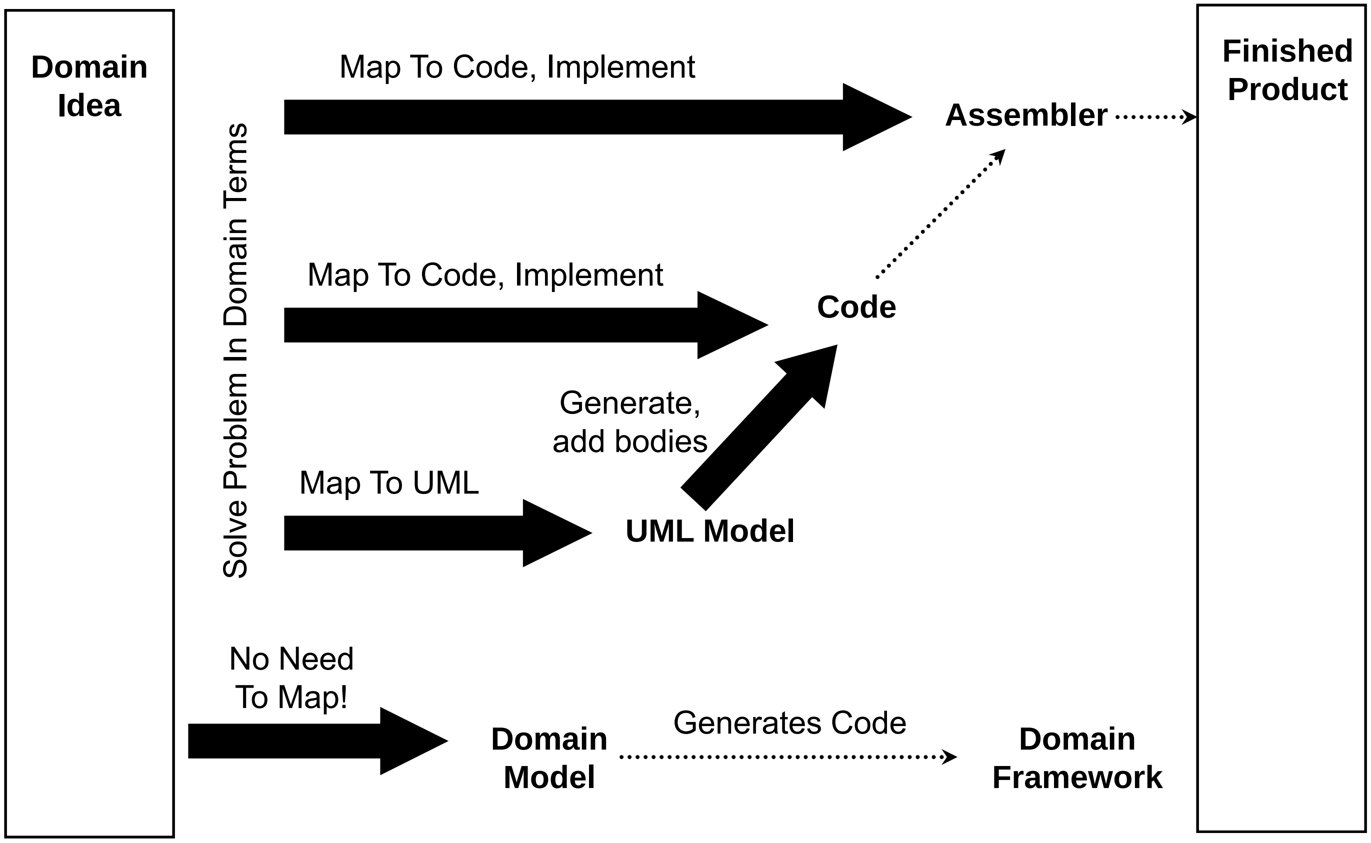
Configure for target platform, application

Design Patterns, Libraries, Etc.

Generate source

Applications

MDSE Raises Level of Abstraction



(Based on Kelly and Tolvanen 2008)

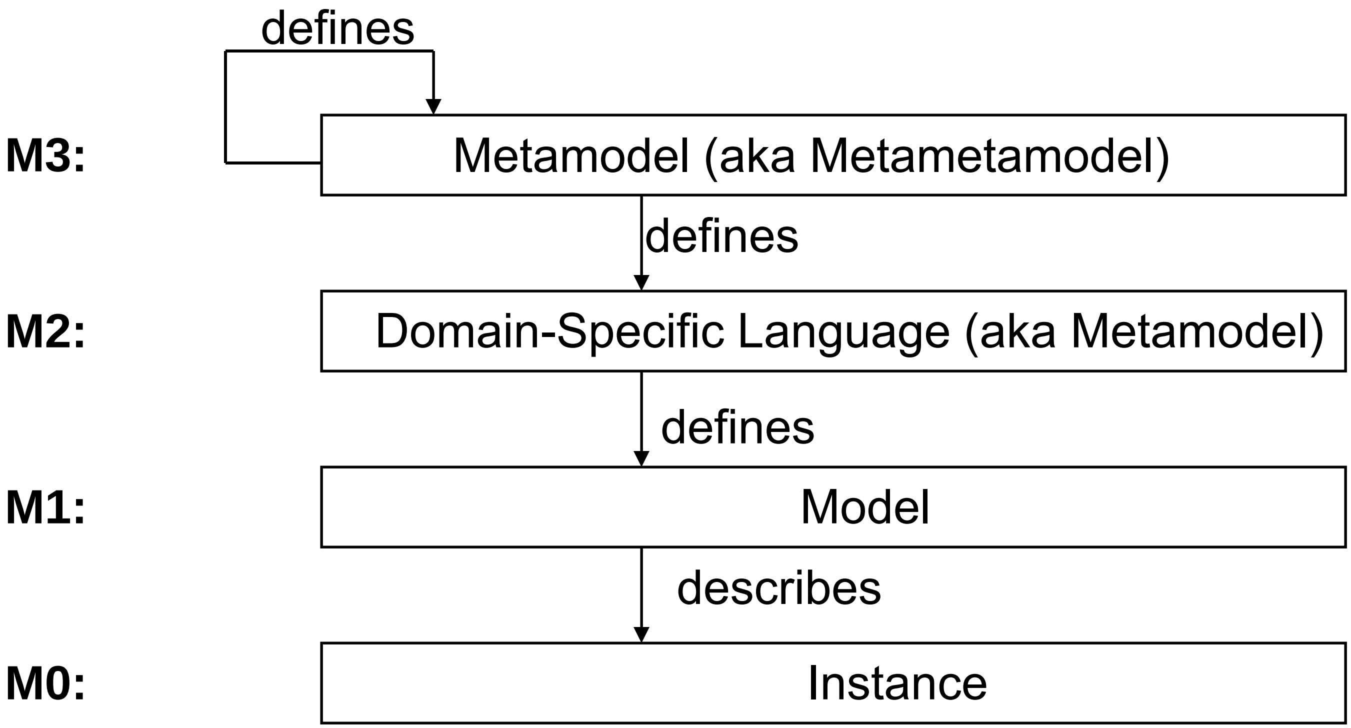
Origins

- Knowledge-Based Software Assistant (KBSA)
  - AFRL project
  - Project meetings became KBS Engineering (KBSE) conference
  - Now IEEE Conference on Automated Software Engineering
- Computer Aided Software Engineering (CASE) tools
  - Often Object-Oriented
  - Often with diagrams for user interaction
  - Functionality: Documentation, prototype simulation, code generation
- Object Management Group (OMG) and Unified Modeling Language (UML)
  - UML created by the “Three amigos”: Grady Booch, Ivar Jacobson, and James Rumbaugh
  - Model-Driven Architecture (MDA) is OMG project

Example MDD Tools

- AndroMDA – OMG MDA-compliant
- ArcStyle - OMG MDA-compliant
- Borland Together
- CA Gen
- CA Plex
- Generic Modeling Environment (GME)
- MetaEdit+
- Oslo
- Rational Software Architect
- Rational Software Modeler
- Telelogic Tau
- Telelogic Rhapsody

Metamodeling Hierarchy



M3: Metamodel (aka Metametamodel)

M2: Domain-Specific Language (aka Metamodel)

M1: Model

M0: Instance

(Based on Stahl and Volter 2006)

SDTATIC
Software Development Tools and Technology Information Clearinghouse

OMG Standards for MDSE

- Model Driven Architecture (MDA)
- MetaObject Facility (MOF)
- Unified Modeling Language (UML 2.0)
- Object Constraint Framework (OCF)
- Query/View/Transformation (QVT)
- XML Metadata Interchange (XMI)
- Common Warehouse Metamodel (CWM) Metadata Interchange Pattern (MIP)

Twelve UML Diagram Types in Three Categories

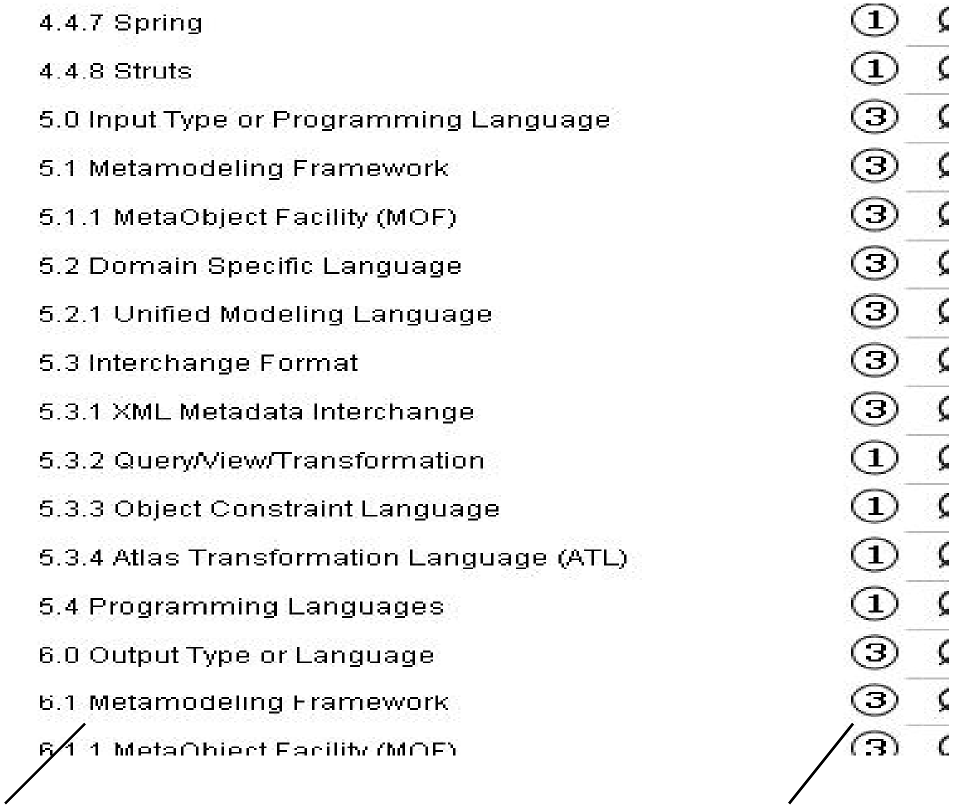
System Structure	**Class**	Classes and their relationships in a logical view of the system
	Object	Objects and their relationships at a specific time
	Component	Organizations and dependencies among software components
	Deployment	Processors, connections between them, and the distribution of components across processors
Model Management	**Package**	Organizes elements of a system into related groups
	Subsystem	Details of a subsystem, including aspects of its operation
	Model	An innovation of UML 2.0

Twelve UML Diagram Types in Three Categories (Cont’d)

System Behavior	**Use Case**	Relationships and the flow of events between actors and a sequence of related transactions
	Sequence	Object interactions in a sequence
	Activity	Flow of control (e.g., business workflow or between methods of a class)
	Collaboration	Object interactions organized around objects and their links
	State Chart	For a given class, states and events that cause a state transition

An *interaction diagram* is a combination of a sequence and a collaboration diagram.

MDD Input Languages Example



4.4.7 Spring	①
4.4.8 Struts	①
5.0 Input Type or Programming Language	③
5.1 Metamodeling Framework	③
5.1.1 MetaObject Facility (MOF)	③
5.2 Domain Specific Language	③
5.2.1 Unified Modeling Language	③
5.3 Interchange Format	③
5.3.1 XML Metadata Interchange	③
5.3.2 Query/View/Transformation	①
5.3.3 Object Constraint Language	①
5.3.4 Atlas Transformation Language (ATL)	①
5.4 Programming Languages	①
6.0 Output Type or Language	③
6.1 Metamodeling Framework	③
6.1.1 MetaObject Facility (MOF)	③

Taxonomy Categories

MDD Importance

SDTATIC
Software Development Tools and Technology Information Clearinghouse

Further Information

Other Suggestions: Q11

SDTATIC Web Site:

http://www.SDTATIC.com/

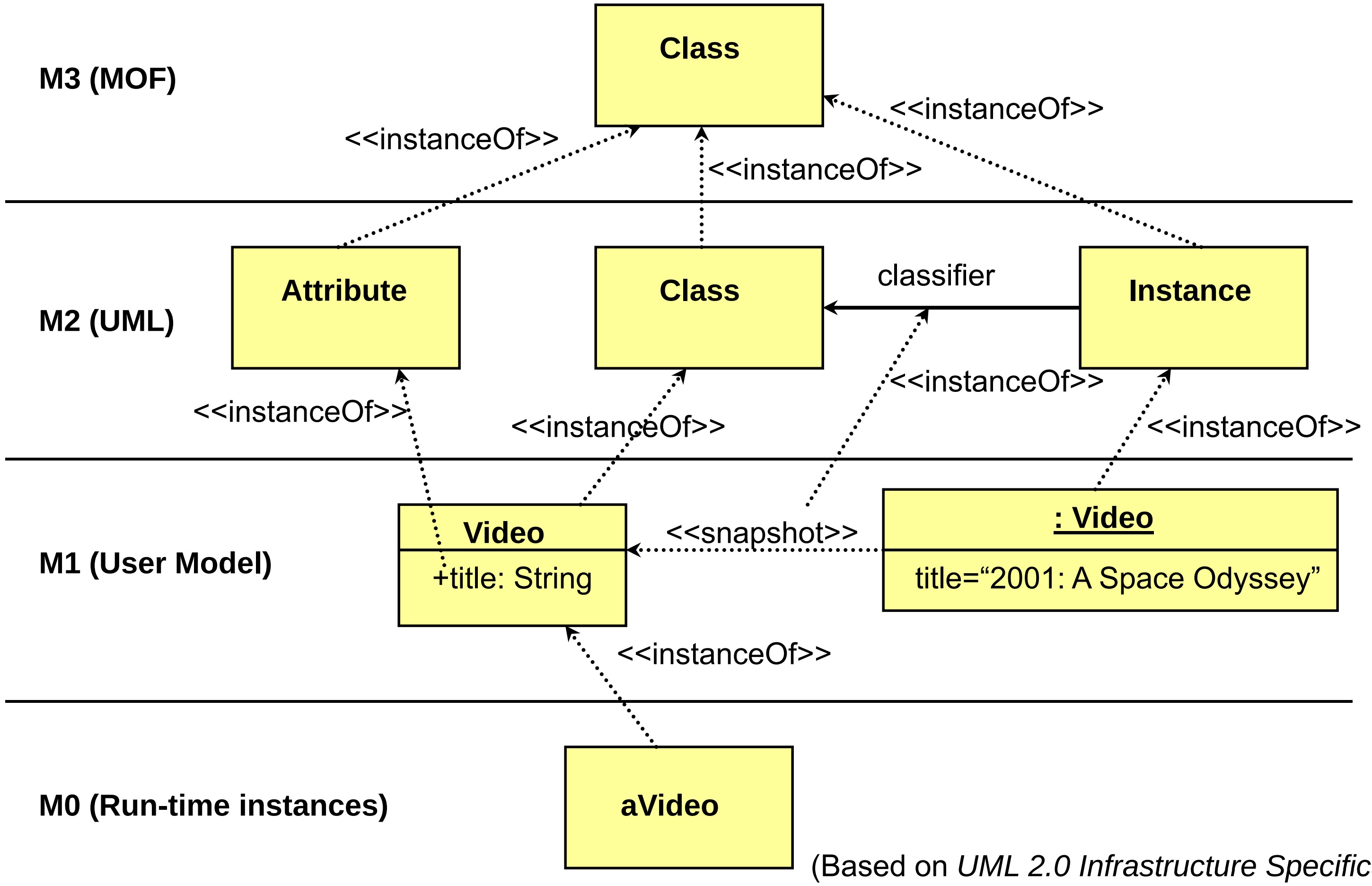
Robert Vienneau
Project Manager
DACS Technical Manager
rob.vienneau@itt.com
315.838.7118

"Just because it's SDTATIC, doesn't mean things don't change"

DACS
DoD Data & Analysis Center for Software

Backup

Metamodel Hierarchy Example



M3 (MOF)

Class

<<instanceOf>>

<<instanceOf>>

<<instanceOf>>

M2 (UML)

Attribute

Class

classifier

Instance

<<instanceOf>>

<<instanceOf>>

<<instanceOf>>

<<instanceOf>>

M1 (User Model)

Video

+title: String

<<snapshot>>

: Video

title="2001: A Space Odyssey"

<<instanceOf>>

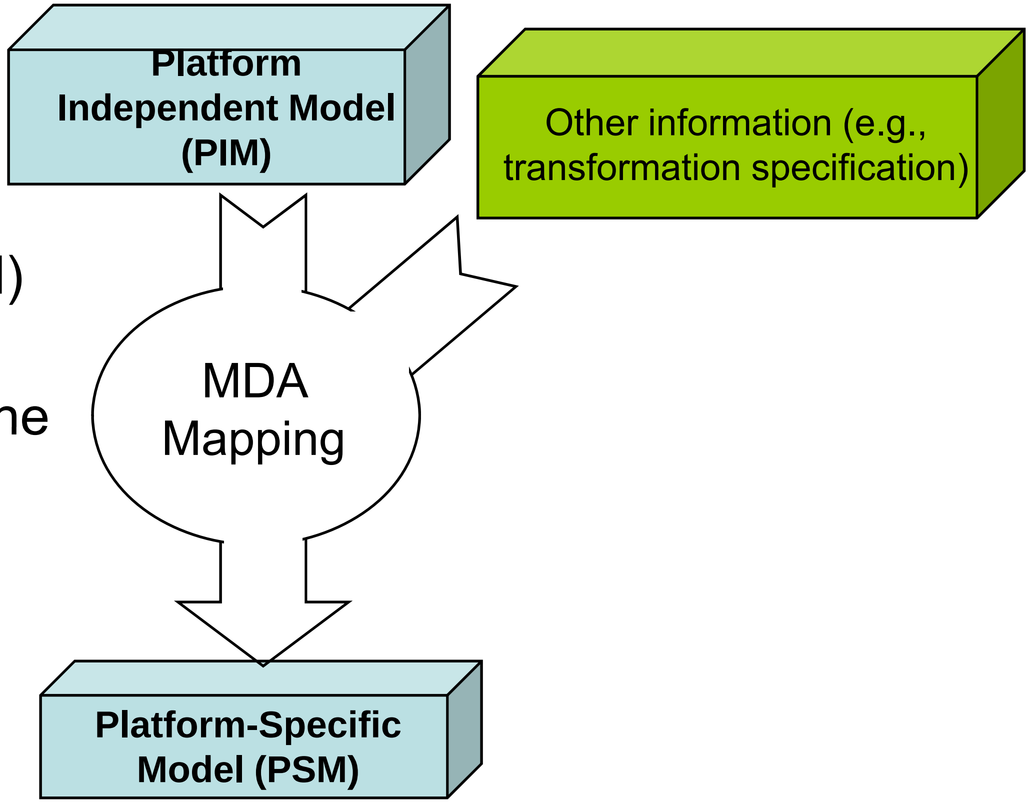
M0 (Run-time instances)

aVideo

(Based on *UML 2.0 Infrastructure Specification* 2003)

OMG Model Driven Architecture (MDA) Process

1. Build the Computational Independent Model (CIM)
2. Build the PIM
3. Transform the PIM into the PSM
4. Generate code from the PSM



Platform Independent Model (PIM)
Other information (e.g., transformation specification)
MDA Mapping
Platform-Specific Model (PSM)